

CONTENTS

1. SetDAG new features.....	3
2. Software configuration	3
2.1. Language choice	3
2.2. Communication parameters choice	3
3. Using the software	5
3.1. Software general working	5
3.2. Race file creation.....	5
3.3. Using an existing race file.....	7
3.4. Creating a number plan	7
3.5. SetDAG operating modes:	9
3.6. Reading the chips.....	10
3.7. Programming bib numbers	11
3.7.1. Contents of the DAGs programming window	12
3.7.2. Chips manual programming	13
3.7.3. Programming through chips barcodes	14
3.7.4. DAG Test programming.....	16
3.7.5. Programming a test bib.....	16
3.7.6. Programming phase various error	17
3.8. Chips checking	18
3.9. Checking the progress of the number plan.....	19
3.9.1. Programming report	21
3.9.2. Complete programming report	22
3.10. Other functions	23
3.10.1. Reader connection test.....	23
3.10.2. Accumulation of the time spent on programming bibs	24
3.10.3. COM port connection/disconnection	24
3.10.4. Security options	25
4.Barcode Programming.....	26
4.1.0. License code	27
4.2.0. register your license	28

History:

Revision	Date	Author	Comments
1	22/03/07	N. MANNINO	Creation
2	04/04/07	N. MANNINO	Modification for version 7.0.1.0
3	29/05/07	N. MANNINO	Modifications after proofreading of S. RUELE & X. MICHEL
4	02/07/07	N. MANNINO	Modification for addition of identified DAG Test
5	08/04/08	X.MICHEL	Modifications regarding system evolution

Document purpose :

This document is relative to the use of SetDAG version 7.0.7.0 to 7.1.0.

1. SetDAG new features

The version 7.0.7 to 7.1.0 of SETDAG enables the following new features:

- Enabling use of V6A and V6B chips (V4 chips still enabled).
- Edition of a programming report allowing to instantly view whether the numbering plan has been respected or not.
- Possibility to encode bib numbers via the barcode reading interfaced with SetDAG.
- Function of countdown of the time spent on programming a race bib numbers.
- Enabling programming of DAGs which validity date has expired.
- Race bibs checking function insuring that all the bibs have been programmed correctly.

2. Software configuration

2.1. Language choice

It is possible to choose SetDAG user interface language. Use the « Language » option in the «Setup» menu. The list of available languages is displayed. A software reboot is required to validate the change.

2.2. Communication parameters choice

In order to communicate with the RFID reader, SetDAG must know the COM port name and the related device communication speed. These parameters are saved so there is no need to input them at each launch of the software, as long as the same equipment is used.

Note for DAGProg users: There are 2 different versions of DAGProg, V12 and V13. Each of these readers creates its own COM port. So, when passing from one DAGProg V12 to V13 (or the other way around), the COM port will have to be changed.

To set the communication parameters, go to « Setup » menu and click « RFID Reader... ». At this point, the window "Communication parameters" will open. It allows inputting the COM port name and its speed.

The communication speed must be kept at 38400.

The list of COM port name only displays ports existing on the PC, it is regularly refreshed.

The "Wait in the field" parameter refers to a waiting time when a chip enters the field of a reader.

This allows to check the physical state of a chip before writing on it. This value is not to be modified except if indicated by DAG System.



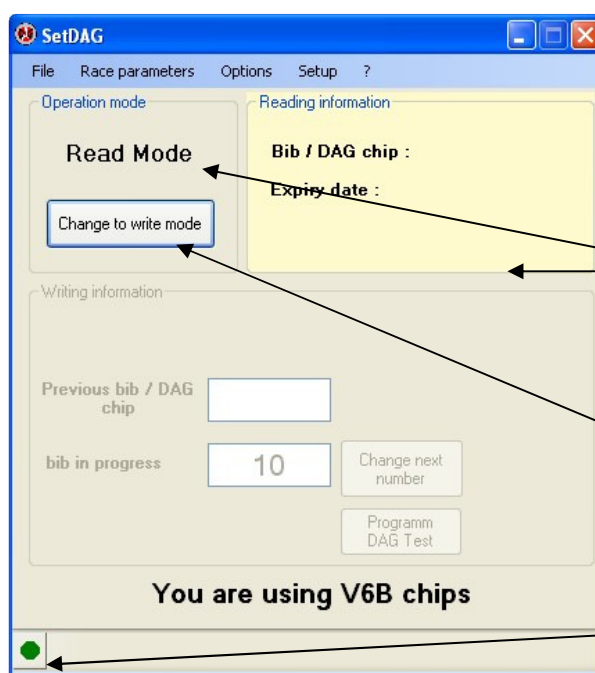
Screenshot 1: COM port parameters choice

Tip: When using a reader connected through USB (DAGProg type), it is possible to set the COM port number it is using from this window. To do so, run SetDAG without having connected the reader. Go to the window "Communication parameters" as described above. Check the proposed ports list. Connect the reader now, wait a few seconds and check the ports list again. The newly appeared port is the one to use with this reader.

3. Using the software

3.1. Software general working

SetDAG allows programming DAG System chips with a bib number chosen by the user. It should always be used with a numbering plan that ensures that every desired bib numbers have actually been programmed.



The software has 3 different working modes: the **read mode**, the **write mode** and the **verification mode** (for V6 chip only).

Indicating the current mode
(text + coloured zone)

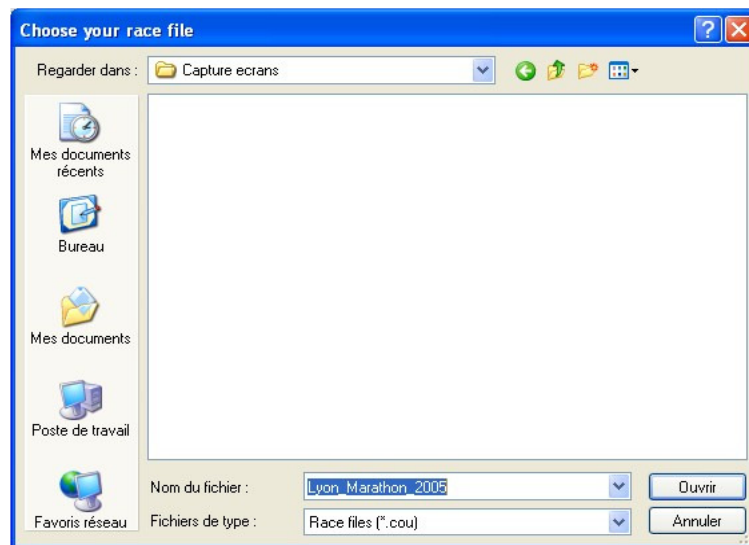
Mode change button

Enter/exit pause mode button

Screenshot 2: Main window of SetDAG
:Read mode

3.2. Race file creation

The first step of chips programming is to create the race for which they will be used. The file race gathers all information on the race and programmed chips. The race file gathers all the information on a race and on the programmed chips. This file should be sent to DAG System before any claims so make sure you keep it in a safe folder.



Screenshot 3: Race file creation

To do so, when SetDAG is launched, a window allowing choosing a race file appears. If a file not existing before is given, SetDAG considers that it's a new race and creates a new blank race file. The race files are automatically created with a ".cou" extension.

The second step is to input the race parameters. All the fields must be filled in: the name of the race, its manager, the date, a client affectation account for the time countdown, the type of chips. Be careful, this step is very important as it will be later impossible to modify the race parameters once the race file is created, i.e., once the OK button has been clicked.

Race information

Name of the race:

Person in charge:

Your DAG System reference:

Date of race: juillet 2007

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
26	25	26	27	28	29	30	1
27	2	3	4	5	6	7	8
28	9	10	11	12	13	14	15
29	16	17	18	19	20	21	22
30	23	24	25	26	27	28	29
31	30	31	1	2	3	4	5

☐ Aujourd'hui : 18/07/2007

Chip type:

☐ V6A (Disponible) ☐ V6B (re-usable) ☐ V4

Screenshot 4: Creating a race file: input race parameters

Race information

Name of the race:

Person in charge:

Your DAG System reference:

Date of race: juillet 2007

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
26	25	26	27	28	29	30	1
27	2	3	4	5	6	7	8
28	9	10	11	12	13	14	15
29	16	17	18	19	20	21	22
30	23	24	25	26	27	28	29
31	30	31	1	2	3	4	5

☐ Aujourd'hui : 18/07/2007

Chip type:

☐ V6A (Disponible) ☒ V6B (re-usable) ☐ V4

Screenshot 5: Creating a race file: all the fields are filled, the race can be created

3.3. Using an existing race file

Launching SetDAG, if an already existing race file is chosen in the race file selecting window, then the relative race information window appears. This one is identical to the window of a race creation except that all the fields are already filled in and in "read only". After validating the window, the programming can resume, the already programmed chips are in the memory.

Click «Back» allows quitting this race and selecting another race file.

Race information

Name of the race: Lyon Marathon

Person in charge: DAG System

Your DAG System reference: RACE 55

Date of race: **juillet 2007**

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
26	25	26	27	28	29	30	1
27	2	3	4	5	6	7	8
28	9	10	11	12	13	14	15
29	16	17	18	19	20	21	22
30	23	24	25	26	27	28	29
31	30	31	1	2	3	4	5

Aujourd'hui : 18/07/2007

Chip type:

☐ V6A (Disposible) ☒ V6B (re-usable) ☐ V4

Back Ok

Screenshot 6: Opening of an existing race file, it is possible to carry on a prior programming session.

3.4. Creating a number plan

A number plan for the bib numbers eases the chips programming procedure by enabling organized separate sets of bibs programming. SetDAG enables using a plan of bib numbers to up of 19 separate programming sets.

This function comes in handy when, for instance, you have bib numbers as such: 1->250, 1000->1250, 2000->2250, 3000->3500, and so on.

	From number	To number	Bib amount
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Total :

OK Cancel

Screenshot 7: Empty number plan

	From number	To number	Bib amount
<input checked="" type="checkbox"/>	1	250	250
<input checked="" type="checkbox"/>	1000	1250	251
<input checked="" type="checkbox"/>	2000	2250	251
<input checked="" type="checkbox"/>	3000	3500	501
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Total : 1253

OK Cancel

Screenshot 8: Valid and in use number plan

In this case, SetDAG can go from one set to the other without the user having to ask to pass to the next programming set. Without this feature, the user would have for instance to ask the software to pass to the 1000 bib once he has finished programming bib number 250, and so forth.

To validate a number plan, use the « Race parameters» menu. To define the start and end values of the numbering sets, tick the boxes on the front of desired line. SetDAG operates a dynamic check on the validity of each set and will notice any inconsistency in red font.

Rules to follow for a numbering plan:

- The sets should not intertwine
- The starting bib number should always be smaller than the ending

number. If it is necessary to code in chips decreasingly (from 1250 to 1000, or 250 to 1 for instance), just program the set 'normally' (from 1 to 250 and 1000 to 1250), then when actually programming the chips, choose "Decrement bib's number"; then finally, fix as current value the maximum value of the set (1250).

Trick: If you have used a numbering plan that you no longer wish to follow, the easiest way is to untick the boxes of each set and click « OK ». This will erase all the different sets.

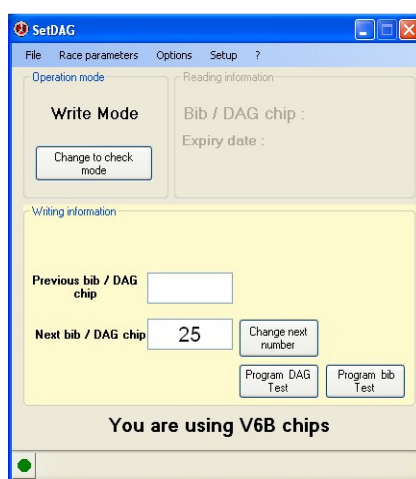
3.5. SetDAG operating modes:

SetDAG has three different operating modes:

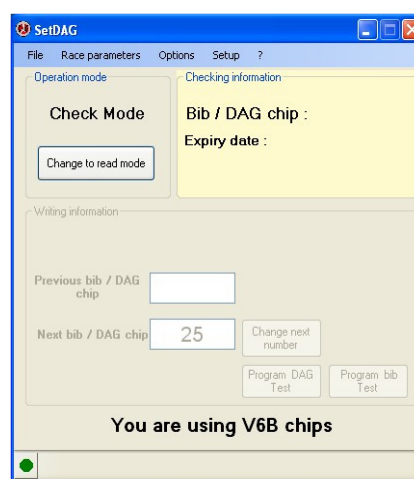
- Read mode: this mode enables reading any DAG presented to the reader and to extract its bib number and validity date.
- Write mode: this mode enables programming the DAG presented to the reader.
- Check mode (V6 programming only) : this mode enables reading the DAG of the current race, and only these ones, so as to insure that they are all properly programmed and ordered. This mode is very similar to the Read mode except on the following features: only the bibs of the race are processed, the read bibs are definitely saved as valid.



Screenshot 9: When starting the application, the Read mode is active, pressing the change mode button enables entering the Write mode.



Screenshot 10: The Write mode is active, any chip presented to the reader is programmed. Pressing the change mode button enables entering the Check mode.



Screenshot 11: The Check Mode is active, the read chips are checked and the order of the numbers is verified. Pressing the change mode button enables returning to Read mode (V6 programming).

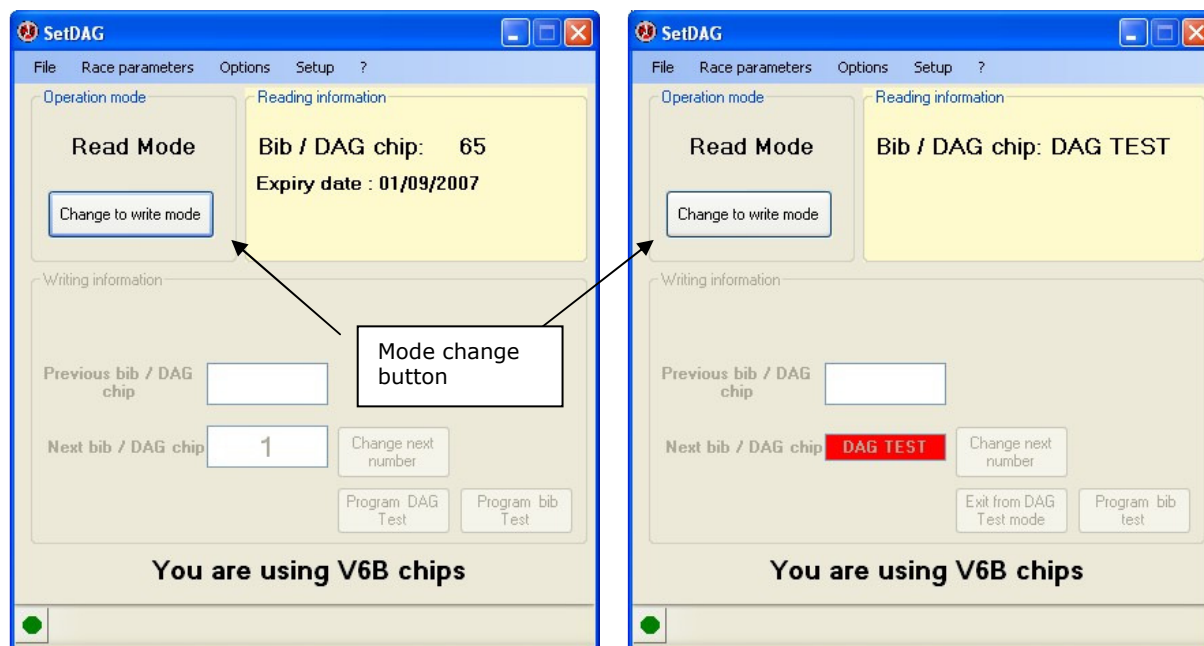
Switching modes is done by pressing the destined button; the different

operating modes can be operated in loop following the order described above.

3.6. Reading the chips

The read mode is active when the frame "Reading information" is colored. To switch modes, use the change mode button.

When in read mode, SetDAG is ready to read all the chips that will be presented to the reader. They are automatically decoded and the bib number and the validity date are displayed.



Screenshot 12: Bib 20912 has been read

Screenshot 13: A DAG Test has been read

The same chip cannot be read twice in a row. After reading a chip, the reader expects a different one. So, whenever a chip seems to have a problem, try reading another chip before trying to read the problematic one.

3.7. Programming bib numbers

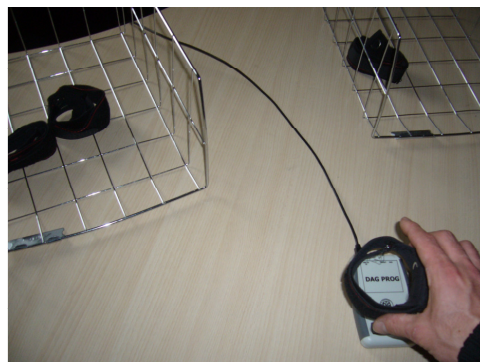
Note on the environmental surroundings during the programming : For optimal security on the outcome of the programming session, DAG System recommends the respect of the following instructions:

- the operator must use only one hand to present the DAG during the programming phase.
- The two piles of bibs (one for the blank bibs, one for the already programmed bibs), must be located on each side of the DAG Prog.
- Each pile should be put in a metallic container in order to prevent any electromagnetic interference that could occur while programming other chips.

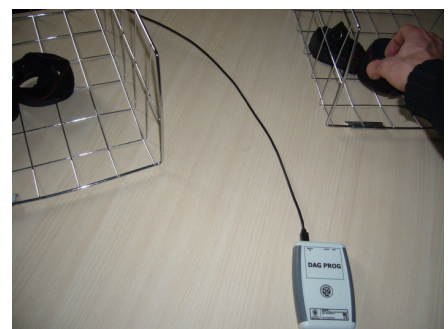
Illustration chip programming:



Screenshot 16: The empty chip are store in a metallic contener.

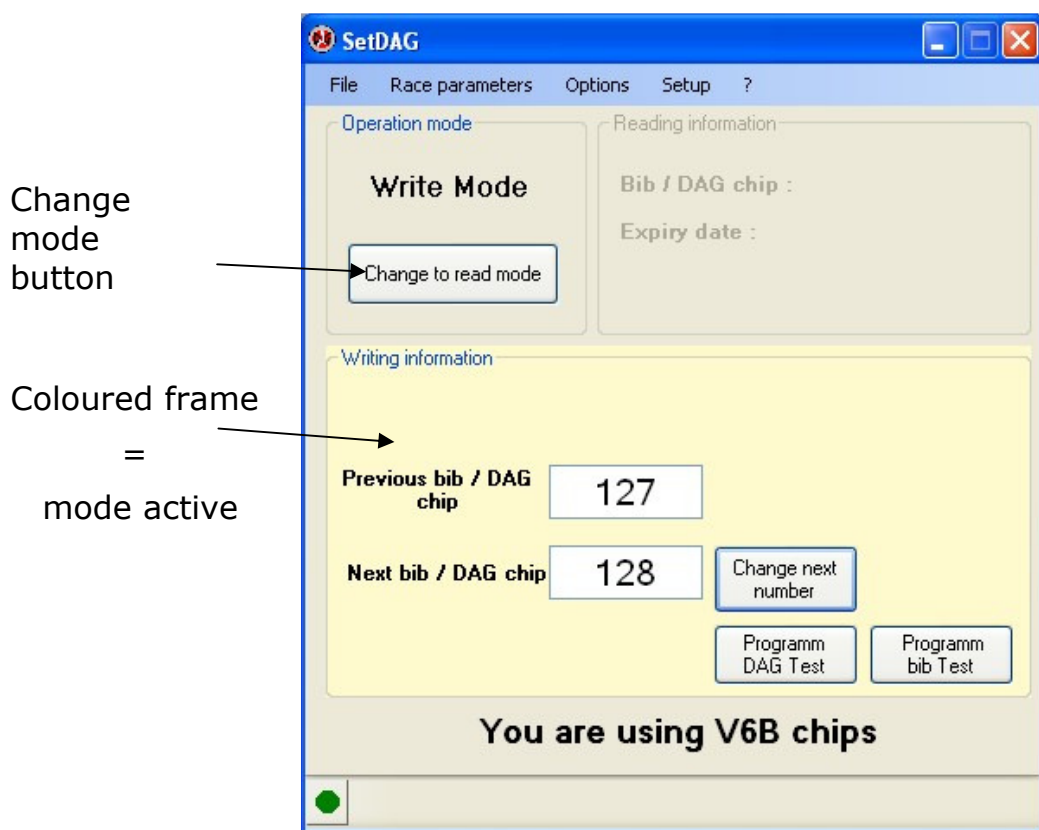


Screenshot 15: the chip is write by the DAGProg.



Screenshot 14: the programmed chip is store into another metallic contener.

To enter the programming mode, use the change mode button. When the « Writing information» frame becomes colored, you are in Programming Mode.



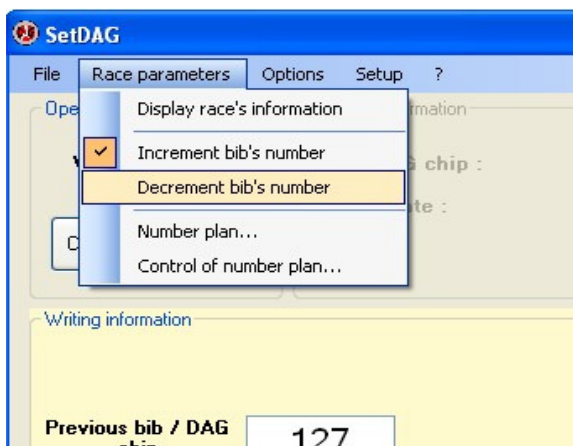
Screenshot 17: SetDAG in « Write mode »

Remark on the V6A chips programming: V6A chips can only be programmed once. We recommend great caution when programming them, as any mistake can cause a loss of the chip. If an already programmed V6A chip is presented to the reader while SetDAG is in write mode, the software will inform the user that the chip has already been written and an error message will appear.

3.7.1. Contents of the DAGs programming window

The window "Writing information" gathers the programming information, it includes:

- **"Next Bib/DAG chip"** field: displays the bib number that will be written when the next chip is placed near the reader.
- **"Previous Bib/DAG hip"** field: displays the last written bib number
- **"Change next number"** button: enables changing the next awaited bib number value.
- **"Program DAG Test"** button : enables changing to DAG Test programming mode.
- **"Program Bib Test"** button : available only when programming V6 chips, it enables changing to bib Test programming mode.



It is also possible to modify the sense in which bibs are consecutively written. Actually, it is possible to write following bib numbers decreasingly to ease operations when the race bib numbers are organized decreasingly. To do so, go to the menu "Race parameters" and choose "Increment Bib's number".

Screenshot 18: Choose the bib numbers programming sense (increasingly or decreasingly)

During the writing operation, a warning light informs the user on the operation evolution:

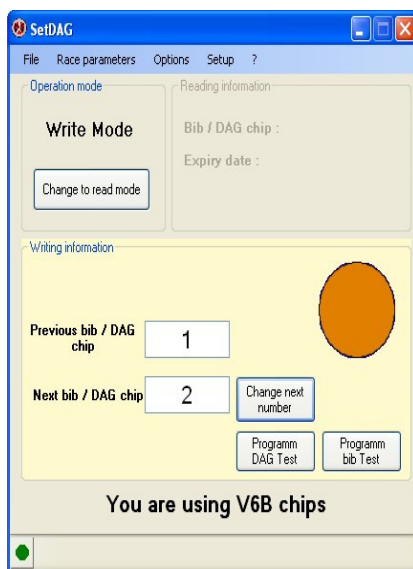
- Orange, after the chip reading, i.e., as the writing process begins.
- Green, at the end of the writing if the procedure went well.
- Red, if an error occurred (an error message explaining the cause of the failure is then displayed).

3.7.2. Chips manual programming

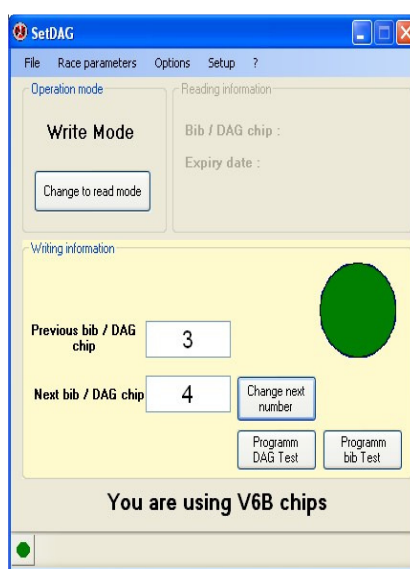
Chips manual programming consists in linear programming following the relative numbering plan. A message appears once the end of the plan has been reached, the writing operation is then over.

To program a chip, present it close to the reader. During each operation SetDAG checks whether the desired bib number hasn't already been

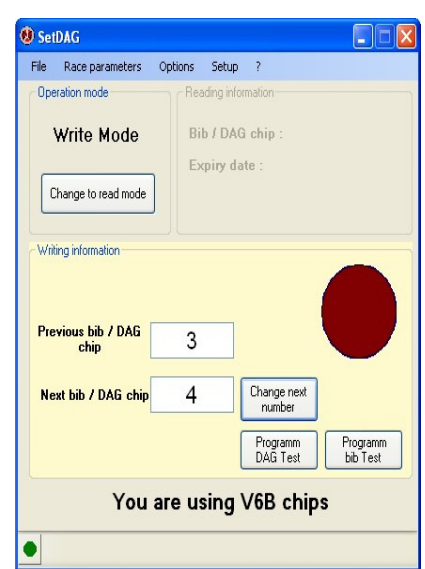
programmed, if it is the case, the operation is cancelled. SetDAG also checks that the chip hasn't already been programmed for this race, if it is the case, a message appears warning the user.



Screenshot 19: Chip writing in progress, the currently writing number is bib 2



Screenshot 21: Chip writing OK, the chip has been programmed as bib 3



Screenshot 20: Writing failed, an error message will inform on the failure reason

A short buzz sound is sent on the PC sound board whenever a chip is successfully programmed.

3.7.3. Programming through chips barcodes

If the bibs have barcodes, it is possible to use them so that SetDAG automatically numbers the chips.

This operating mode is a particular case of the writing mode. It imposes a peculiar behavior on SetDAG, making the software become predominant, i.e., it becomes almost impossible to use other applications on your computer at the same time.

To enter this mode, press the function key 'F5'. The zone displaying the current bib appears in blue to ensure that it is the barcode programming mode. In this mode, the chips are programmed once the barcode is read. To quit this mode, press 'F5' again to resume to the standard mode.

Important notice on the barcode mode use :

When the barcode mode is active, be sure to check that the cursor is always in the text zone containing the current bib, otherwise, the barcodes detection will not work.

3.7.4. DAG Test programming

Programming DAG Test is a particular function of the Write mode. By pressing the 'Program DAG Test' button, you enter the DAG Test mode and can create as many as you want. Then, clicking on that button again, you get back to the standard writing mode.

3.7.5. Programming a test bib

This function is only available when programming V6 chips. This operation has the same functioning as programming DAG Test. This mode programs test bibs that are identical to those of a race. They are used to test the continuity of the field.



Enter/Exit test bib programming button

Screenshot 22: Programming test bib in progress

3.7.6. Programming phase various error

During chip programming, various error or information messages can appear, such as:

- "No chip in reader's field.": The chip has been taken away from the reader's field too quickly and the writing process couldn't be completed. The chip content is undetermined.
- "Bib number already done.": The bib number you want to write in that chip was already used for that race. It is thus impossible to write this number again and SetDAG passes to the next number to write. To write in again this number all the same, you will need to pas in 'unsecured' mode by entering the appropriate code in the 'Security' window of the "Configuration" menu.
- "Chip is already programmed." : The chip presented to the reader has

already been programmed for this race. The bib number previously written in the chip appears in a message. It is possible to write in this chip again (the previous number written in the chop will be nullified), or simply cancel the operation and carry on with another chip.

- "No chip found" (displayed in barcode mode only): a barcode has been read but not chip was found by the reader.

3.8. Chips checking

This operation enables validating a race's bibs programming session, **it only concerns V6 chips**. Although not compulsory for using chips, DAG System recommends this mode in order to avoid any bib number inversion ; not or wrongly programmed chips problems.

Bibs shall be checked accordingly to the numbering plan (either in growing or descending order).

To check the bibs, enter Check mode. This operating mode is identical to the Read mode except for the following points:

- The bibs that are not part of the race are not displayed and are pointed out to the user so as to take them out.
- Bibs's reading is done following numbering order. If there is a leap in the planned numbering, the user is informed.
- Each bib from the race is checked and is then saved as valid in the SetDAG database.

For the checking, just as for programming, it is possible to choose the direction of scanning through the bib numbers. In check mode, in the menu 'Race parameters' you can find the fields: 'Check by incrementing' and 'Check by decrementing'.

The checking results appear in the 'Numbering plan verification' window, accessible through the 'Race parameters' menu.

Important note on checking:

After checking the bibs, DAG System recommends that you package the checked bibs and stocked them far from the workstation, so as to prevent any accidental programming.

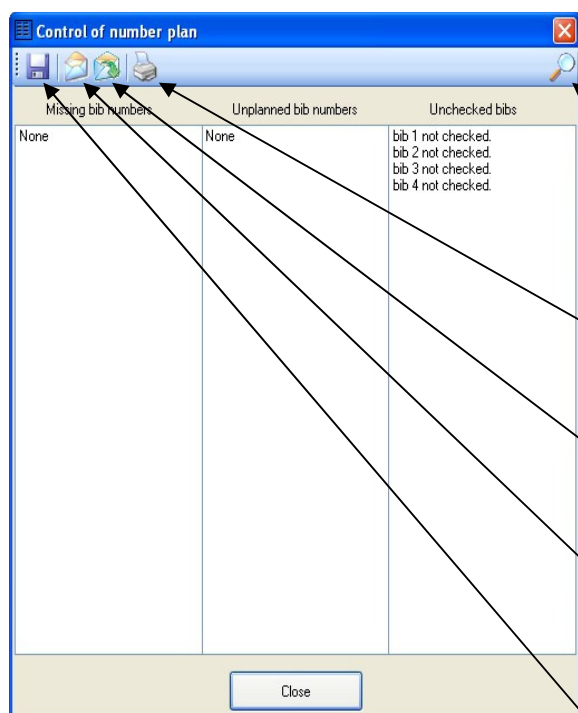
The different possible situations when checking a DAG:

- The checked DAG is valid and respects the numbering. Its number is displayed in the verification window.
- The checked DAG is invalid: a message warns the user. This DAG is to be reprogrammed since it's not usable on that race. Note: to reprogram, it might be necessary to enter 'Unsecured' mode. Please refer to the chapter on secured modes.
- The checked DAG is not part of this race: a message warns that the DAG is not programmed for that race, it has to be either reprogrammed or taken away from this race chips.
- The checked DAG is valid for this race, but the order of passage doesn't match the numbering plan. In check mode, SetDAG checks the order in which the DAG are presented so as to avoid mixing between two consecutive bib numbers. A message will warn you that the presented bib is not the one that should normally follow the last checked number. In this case, 2 possibilities: cancel the passage of this DAG so as to pass to the next expected number and thus arrange the bibs, or, carry on with the checking by making a 'leap' in the numbering.

3.9. Checking the progress of the number plan

SetDAG allows checking that the fixed numbering plan was thoroughly followed. To do so, in the "Race parameters" menu can be found "Control of number plan...", it allows opening the verification window.

This window allows spotting the bibs that haven't been programmed yet, or bibs that had been programmed while they weren't planned for the race. It allows spotting bibs which are not been checked too.



Displays the "complete programming report"

Print the programming report

Send the programming report and race file by email

Send the programming report by email

Save the programming report on text file format on the hard drive.

Screenshot 23: Checking that all bibs planned has been programmed but 4 has not been checked at now.

Contents of each column :

- The first list enables visualizing the bibs that have been planned in the numbering plan but that have not been programmed yet.
- The second list displays the bibs that have been programmed, but that are not in the numbering plan.
- The third list displays the list of bibs contained in the numbering plan that have not been checked yet (this column is meaningless if your are not using V6 chip).

At the end of the programming of the bibs or DAG Chips of a race, the 3 list should be empty, meaning that:

- All the planned bibs in the numbering plan have been programmed (first list).
- All the programmed bibs have been checked.
- No additional bib (not planned) has been kept.

Remarks: To view the complete list of the written bibs, you must ask for the 'programming report'.

3.9.1. Programming report

The programming report is a sum up version of the numbering verification window and of the race information. It is possible to print the report, to save it on the hard drive or to send it by email.

When saving on the hard drive, the report is automatically saved in the same folder as the race file; it has the same name except that it has a '.doc' extension.

To send it by email, 2 options: sending the report alone or along with the race file (*.cou). During the sending process, an email address is asked for the report's addressee. It is possible to add others addressees by separating the email addresses with ';'.

Information contained in a race report:

- Name of the race
- Number of programmed bibs
- The repartition, based on the programming method: manual = bib number fixed by SetDAG, according to the numbering plan; barcode = bib number fixed by a barcode value.
- Type of coded chip for this race: V4 or V6B
- Race file creation date
- Name of the person in charge
- SetDAG software version
- Time spent on programming these bibs
- The client affectation account
- Name and date of race
- Numbering plan
- List of the bibs that have not been checked (only for V6 programming).
- The order followed during checking (only for V6 programming).

Race report example:

DAG SYSTEM PROGRAMMING REPORT

test.cou
58 bibs in file
Programming mode used : 58 manual ; 0 bar code
Kind of chip : V6B
Creation date : 7/31/2007
Responsible's name : Nicolas
Version : SetDag v.7.0.6.0
Elapsed time for this file : 0h40
Account reference : test
Race : Test BDD 5, 8/31/2007

Number plan : 58 Bibs

1 -> 29
51 -> 79

Missing bib numbers : 0

Unplanned bib numbers : 0

Unchecked bibs : 4

List of missing bib numbers :

None

List of unplanned bib numbers :

None

List of unchecked bibs :

1, 2, 3, 4

List of checking range :

5->29, 79->51

C:\SVN_Local\SetDAG V7\test.cou

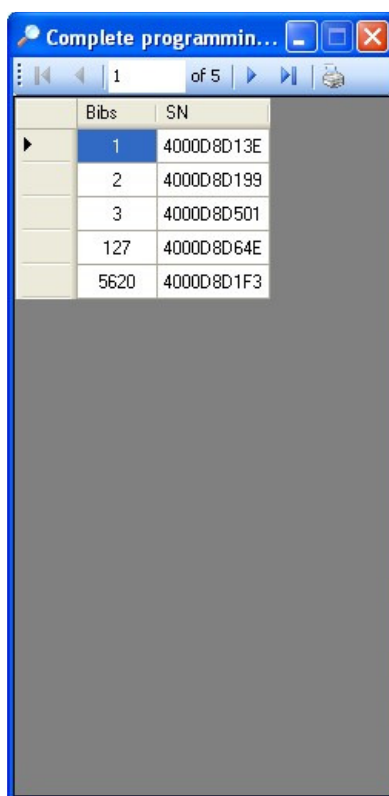
Done at : Thursday, August 02, 2007 to 10:41:02 AM

Note : The programming report can be in a different form whether it is printed from SetDAG or exported on a different file. However, the given information remains the same.

3.9.2. Complete programming report

The race report is the complete list of the programmed chips' serial numbers, as well as their associated bib numbers.

It is displayed by clicking the magnifying glass shaped icon in the verification window. It is possible to print the content of this file by clicking the printing icon.



	Bibs	SN
▶	1	4000D8D13E
	2	4000D8D199
	3	4000D8D501
	127	4000D8D64E
	5620	4000D8D1F3

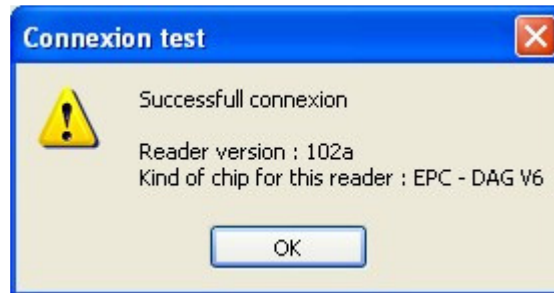
Screenshot 24: Race report example

3.10. Other functions

3.10.1. Reader connection test

It is possible to check the reader with which the software communicates in order to check on its version and if it operates well.

To do so, use the "Test connexion with reader" option, in the "Setup" menu.



Screenshot 25: Succeeded communication test with the reader

The following information is displayed:

- Reader software version
- Type of chips that the reader can read, either V4 or V6.

3.10.2. Accumulation of the time spent on programming bibs

SetDAG performs a countdown of the time spent on each race and this time is displayed in the programming report. The software counts all the periods when the race file has remained opened. So, in order not to distort the measure, it's possible to stop the countdown to have a break or to do another task. To do so, enter in Pause mode.

The pause mode is activated through various ways:

- From the "Options" menu
- Clicking the green button at the bottom of the soft window.
- Pressing the 'F11' key.

The pause mode is deactivated by the same actions.

The pause mode stops the countdown, but it also prevents any operation on the chips and the software frees the communication port.

3.10.3. COM port connection/disconnection

It's possible to use several software at the same time as SetDAG (except in Write mode with the barcode option activated). These softwares can need the COM port used by SetDAG. It is possible to free the port without closing SetDAG and then having to re-open it once the others tasks are done on the computer. To do so, the 'Options' menu proposes the following functions:

- Connect to reader: The COM port is used exclusively by SetDAG.
- Disconnect to reader: SetDAG frees the COM port.

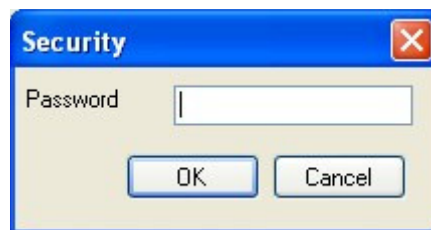
- Read the connection to reader status: displays whether the COM port is used by SetDAG or not.

Of course, if the COM port is already being used by other software when you choose 'connect to reader', SetDAG will not be able to access the port and an error message will appear.

3.10.4. Security options

SetDAG enables the user to identify himself in order to be granted access to secured operations.

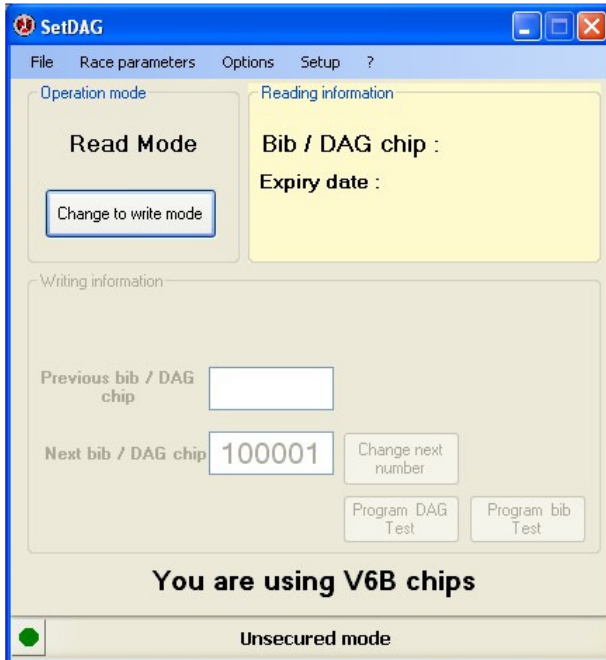
- Unsecured mode: this mode allows programming a bib number more than once for the same race. SetDAG forbids programming twice the same bib numbers. However, in case of programming mistake (skipping a bib number for instance), it can become necessary to reprogram already written bib numbers. To do so, enter the unsecured mode by:
 - In the "Configuration" menu, click "Security", this opens the following window:



Screenshot 26: Identification window to access extended rights.

In the password field, enter: 'DAG_070'.

- The software will inform you that the password is correct. If not, try again.
- Once in 'unsecured' mode, the main window confirms it in the status bar.



Screenshot 27: *SetDAG* is in *Unsecured mode*, pay extra attention to every executed operations and displayed messages.

Once in this mode, SetDAG allows duplicate bib numbers. We advise not to stay in this mode too long, in order to ensure a better security regarding the respect of your numbering plan.

To quit this mode, repeat the same operations done to access it.

We advise you to execute the 'unsecured' operations all in once, at the end of the programming process. To do so, when programming, keep aside all the problematic DAGs.

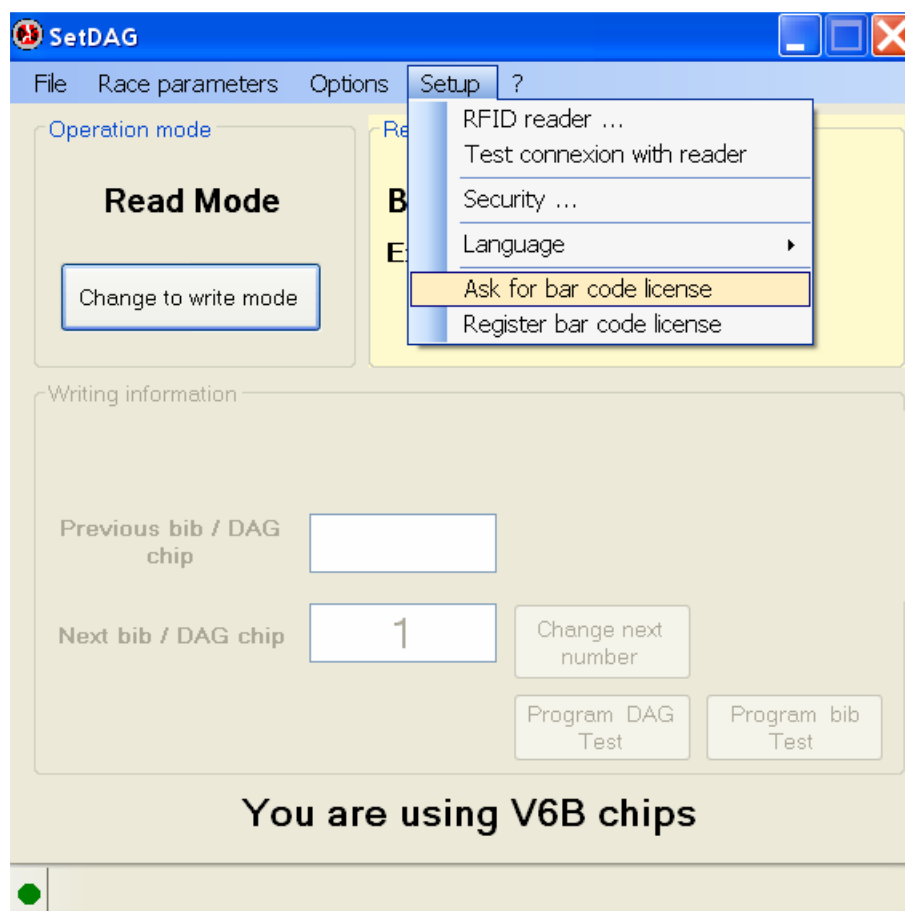
4. Bar Code Licence

4.1. Licence code

In order to use this option you must obtain a user's Licence

Please send to DAG System an official demand + Bar Code Key

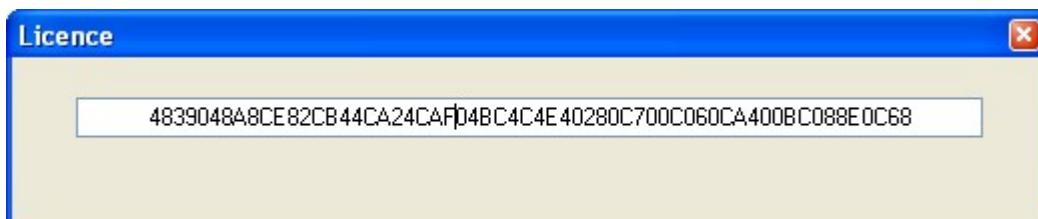
How to get Bar code Key. Select "Configuration" and then "Ask for Bar Code license".



If you are already login, an error message appears



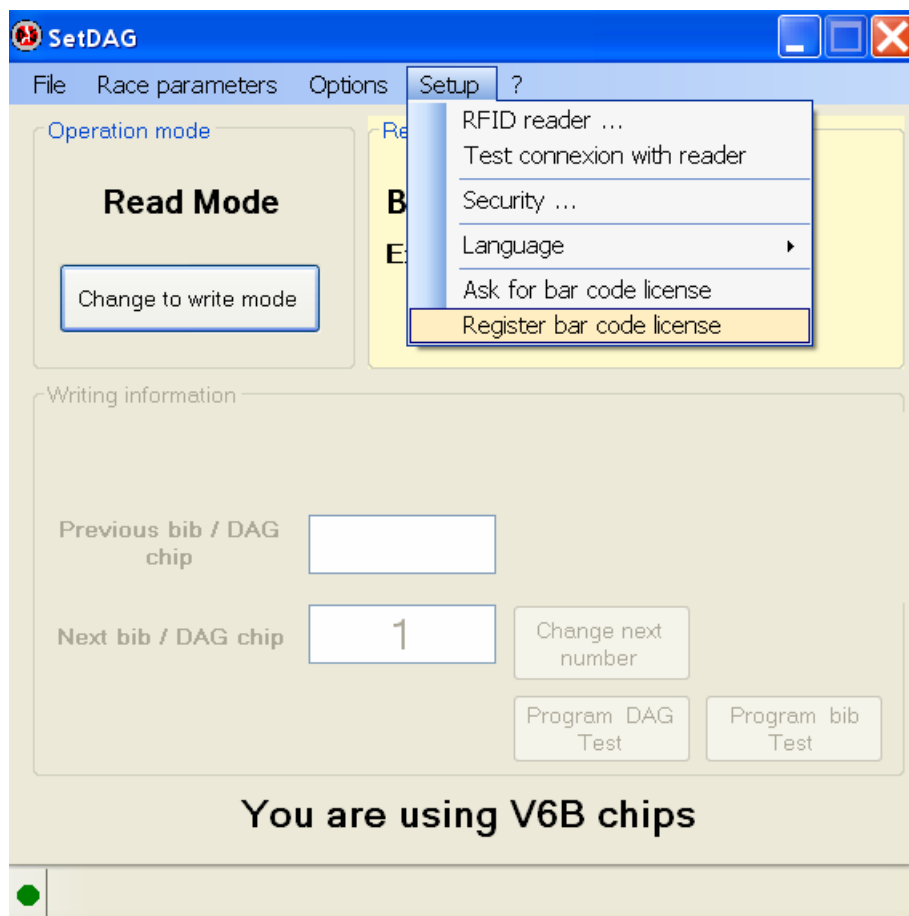
For every new login the code bar key appears



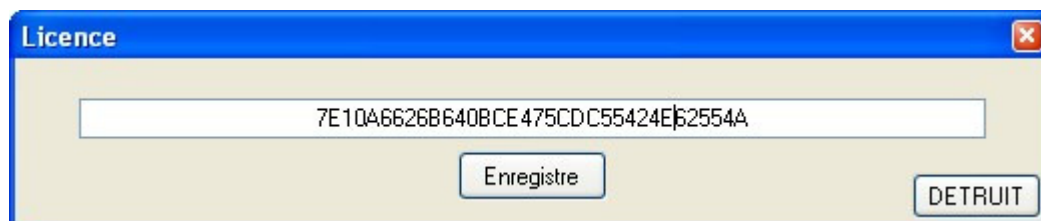
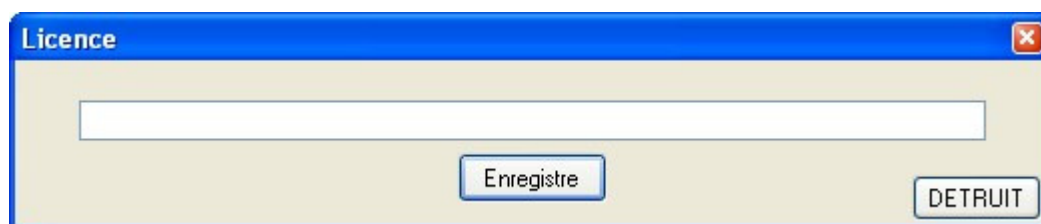
Send this code to DAG system.

4.2. Register your licence

Select "Configuration" and then "register bar code license".



Copie the DAG System code in this windows



Click on « Register »

If any error in the code this message appears.



Verify your licence and contact DAG System

Licence correct = this message appears.



You are now ready to use the Bar code programming process.

Disclaimer:

DAG System reserves the right to make changes, without notice, to any product herein to improve reliability, functionality, or design. DAG System advises its customers to obtain the latest version of device data sheets to verify, before placing orders, that the information being relied upon by the customer is current.

Information furnished by DAG System is believed to be accurate and reliable. However, DAG System does not assume any liability resulting from the application or use of any product described within.

DAG System' products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and DAG System board.

Life support devices or systems are devices or systems that (a) are intended for surgical implant to the body or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Other brand and product names may be registered trademarks or trademarks of their respective holders.